

IN THE CLAIMS:

Please amend the claims as follows:

1-9. (Canceled)

10. (Currently Amended) A method of providing a user access to functional modules from within an application, used to build queries, issue queries and/or, view query results, during a query session comprising:

assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database, wherein the assigned metadata requirements specify conditions required for successful execution of the functional module, wherein at least one condition defines at least one user role required for successful execution of the functional module;

collecting runtime metadata relating to a query session one or more result fields of a query, wherein the metadata is collected after the composition of [[a]] the query;

obtaining a list of functional modules that are accessible from within an application used during the query session;

identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements; and

providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute.

11. (Original) The method of claim 10, wherein the runtime metadata comprises attributes of fields involved in a query or query results.

12. (Original) The method of claim 10, wherein the runtime metadata comprises content contained in query results.

13. (Canceled)
14. (Previously Presented) The method of claim 10, wherein obtaining metadata associated with the functional module comprises examining a signature validation.
15. (Previously Presented) The method of claim 10, wherein the metadata associated with at least one of the functional modules comprises at least one of: one or more input parameters required for successful execution of the functional module, one or more output parameters required for successful execution of the functional module, and a security credential required to execute the functional module.
16. (Original) The method of claim 10, wherein at least one of the functional modules analyzes query results.
17. (Original) The method of claim 16, wherein:
the runtime metadata comprises the names of fields in a result set; and
the limited subset of functional modules comprises functional modules requiring data from fields in the result set as inputs.
18. (Original) The method of claim 10, wherein the runtime metadata comprises information related to a query building session.
19. (Original) The method of claim 18, wherein:
the information related to the query building session comprises a specified focus of the query; and
identifying a limited subset of the functional modules that will successfully execute comprises identifying functional modules associated with the specified focus.

20. (Currently Amended) A computer readable storage medium containing a program which, when executed, performs operations for providing a user access to functional modules from within an application, comprising:

assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database, wherein the assigned metadata requirements specify conditions required for successful execution of the functional module, wherein at least one condition defines at least one user role required for successful execution of the functional module;

collecting runtime metadata relating to a query session one or more result fields of a query;

obtaining a list of functional modules accessible from within the application

identifying a limited subset of functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements; and

providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute.

21. (Currently Amended) The computer readable storage medium of claim 20 wherein the application is a query building application.

22-26. (Cancelled)

27. (Currently Amended) A data processing system for providing a user access to functional modules from within an application comprising:

a data repository;

a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules, wherein at least one condition defines at least one user role required for successful execution of the functional modules;

an application from which the functional modules are accessible, wherein the application is configured to:

collect runtime metadata related to one or more result fields of a query after the composition of [[a]] the query; and

present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules.

28. (Original) The data processing system of claim 27, wherein the data repository comprises XML data structures used to store runtime metadata.

29. (Original) The data processing system of claim 27, wherein the data repository comprises relational database tables used to store runtime metadata.

Please add the following new claims:

30. (New) The method of claim 10, wherein the runtime metadata relating to one or more result fields of the query comprises one or more of:

a result field name;
a data type for the result field; and
a number of records returned.

31. (New) The computer readable storage medium of claim 20, wherein the runtime metadata relating to one or more result fields of the query comprises one or more of:

a result field name;
a data type for the result field; and
a number of records returned.

32. (New) The data processing system of claim 27, wherein the runtime metadata related to one or more result fields of the query comprises one or more of:
a result field name;
a data type for the result field; and
a number of records returned.